Learning objectives

- Making scripts
- Understanding compute clusters
- Using Slurm & troubleshooting SBATCH scripts
- Transferring and downloading files

Make a script

- In a terminal on your local machine:
- Open a new file called `test_script.sh`
- Add the following four lines:

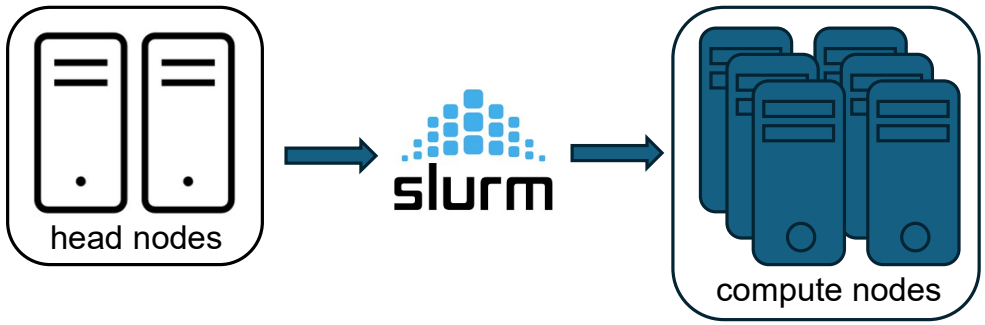
```
#!/bin/bash
pwd
sleep 3
echo “This is a test script”
```

- Save and exit, then run the script with:

```
sh test_script.sh
```

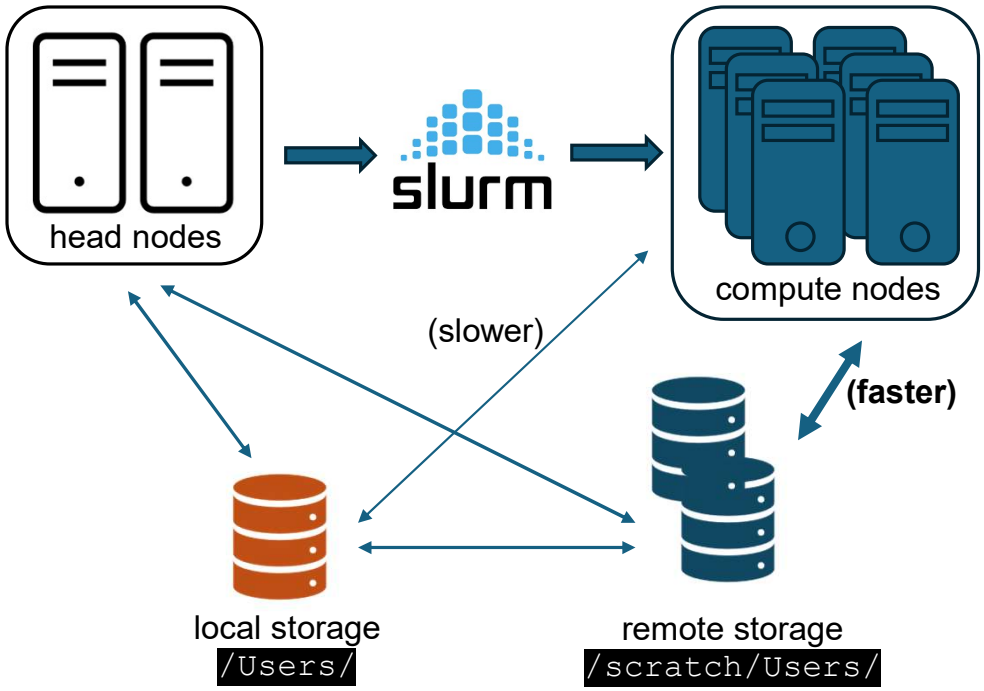
Compute cluster architecture

- Head Nodes
  - Limited resources
  - Intended for basic interfacing, commands, writing and submitting scripts.
- Compute Nodes
  - Extensive resources
  - Requires job scheduler to manage resources
  - Receives Slurm job scripts and executes your commands



Compute cluster architecture

- Home Directory storage
  - Off-site from cluster (slow)
  - Snapshots & replication
  - Typically located: `/Users/<username>`
- Scratch Directory storage
  - On-site with cluster (fast)
  - NOT backed up, $$$
  - Typically located: `/scratch/Users/<username>`



A few Slurm Commands

```
$ sinfo                          # List compute node information
$ squeue                         # List compute node queue
$ squeue -u <username>           # List your compute node queue
$ squeue --me
$ sbatch <script>.sbatch         # Submit sbatch script to queue
```

SBATCH header

```
#!/bin/bash                                                    #shebang, instructs computer to use bash
#SBATCH --job-name=downloadfastq                                        # descriptive name for job
#SBATCH --mail-user=you@email.com                                             # where to send mail
#SBATCH --mail-type=FAIL                                  # email options ALL, BEGIN, END, FAIL, NONE
#SBATCH --time=00:10:00                                                 # Time limit hrs:min:sec
#SBATCH --partition=short                                                     # slurm partition
#SBATCH --nodes=1                                       # number of nodes to use, should typically be 1
#SBATCH --ntasks=1                                         # how many processors are needed for the job
#SBATCH --mem=256mb                  # the amount of memory (RAM) to use, formatted 256mb, 8gb, etc.
#SBATCH --output=/scratch/Users/<username>/eofiles/%x_%j.out          # path to output file location
#SBATCH --error=/scratch/Users/<username>/eofiles/%x_%j.err           # path to error file location
```

Output/error files are primarily used for troubleshooting:

- These files capture STDERR and STDOUT that isn't captured within the script
- %x and %j are Slurm variables that will be **automatically** filled into the filenames
- %x is the job name. If not specified in the header, it will be the name of your sbatch script file
- %j is the unique job ID assigned to your job by Slurm
- Using these variables ensures that you'll always have unique filenames for every job

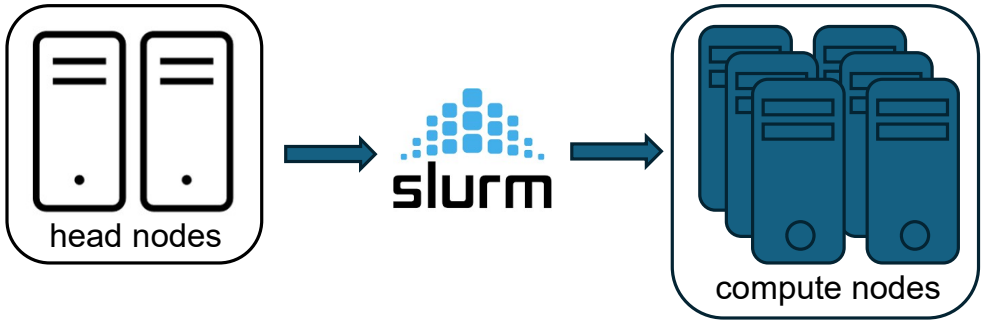
SBATCH header

```
#!/bin/bash                                                    #shebang, instructs computer to use bash
#SBATCH --job-name=downloadfastq                                        # descriptive name for job
#SBATCH --mail-user=you@email.com                                           # where to send mail
#SBATCH --mail-type=FAIL                                  # email options ALL, BEGIN, END, FAIL, NONE
#SBATCH --time=00:10:00                                                  # Time limit hrs:min:sec
#SBATCH --partition=short                                                       # slurm partition
#SBATCH --nodes=1                                       # number of nodes to use, should typically be 1
#SBATCH --ntasks=1                                         # how many processors are needed for the job
#SBATCH --mem=256mb                  # the amount of memory (RAM) to use, formatted 256mb, 8gb, etc.
#SBATCH --output=/scratch/Users/<username>/eofiles/%x_%j.out          # path to output file location
#SBATCH --error=/scratch/Users/<username>/eofiles/%x_%j.err            # path to error file location
```

- How to know how many ntasks, memory, and time to request?

Run an SBATCH job!

- Day 3 worksheet part 1
  - Create an SBATCH script with a header and a command
  - Run the script as a job
  - Troubleshoot



How to know when a job fails

- Expected files are empty or missing
- Job finishes much quicker than expected
- Errors appear in your error file

Why does a job fail?

- TYPOS
- Incorrect command/parameters
- Incorrect paths
- Incorrect SBATCH header
- Incorrect file format
- Issues with software versions

Understanding modules

- Environmental variables alter how we interact with the cluster or help us “find” commands
  - An important environmental variable is `$PATH`, which specifies where the computer looks for commands (look at it with `echo $PATH`)
- Modules are set up by admin and allow you to easily change environmental variables
  - Not present on every compute cluster
  - Modules are automatically unloaded after a session terminates

Module Commands

```
$ module avail                          # List available modules
$ module spider      # Describe modules (with tab complete)
$ module load <module>                  # Load specific module
$ module list                  #  List currently loaded modules
$ module unload <module>            # Unload specific module
$ module purge                       # Unload all current modules
```

Analyze data!

Day 3 worksheet Part 2

- In your SBATCH script, write a command to run the program FastQC
- Run the script as a job
- Troubleshoot if necessary
- Transfer the HTML output to your local computer and view it

Homework for day 4

- Videos for day 3 (if not already done)
- Videos for day 4
- Homework_day3.md worksheet